Drone trajectory control with learning based system identification

Antoine Mahé, Matthieu Geist, Cédric Pradalier

CentraleSupelec, Université de Loraine, CNRS, LORIA, UMI2958 GT-CNRS, LIEC

antoine-robin.mahe@centralesupelec.com

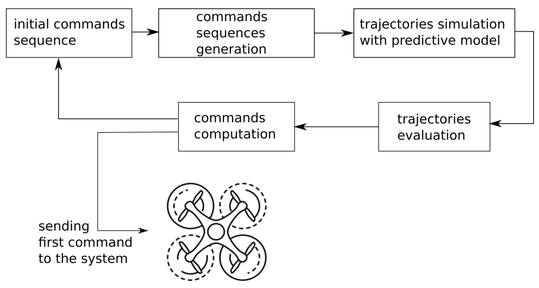
Introduction

Developing aerial robotic in the Greater Region is the focus of the GRoNe project (FEDER INTERREG VA). As part of such a project my thesis subject "Reinforcement learning for aerial robotic" aims at exploring the complementarity between the inspiration from automation [3] and more recent machine learning approach [5].

Control algorithm

First approach to the control problem: implementation of state of the art Model Predictive Control (MPC) controller. Then we used Reinforcement Learning framework to improve from them.

MPPI



We use here a version of MPC developed in [4], Model Predictive Path Integral (MPPI).

This controller rely on a dynamic model of the system to compute an optimal command.

Figure 1: Sampling of trajectories for a mobile robot

System modeling

MPC needs a model of the system. One way to make such a model is to do system identification.

Neural Network approach

System identification can be done through standard algorithms such as ARX [1].

Machine learning algorithms are also very powerful to achieve system modeling.

We will look at the advantages and disadvantages of such neural network model.

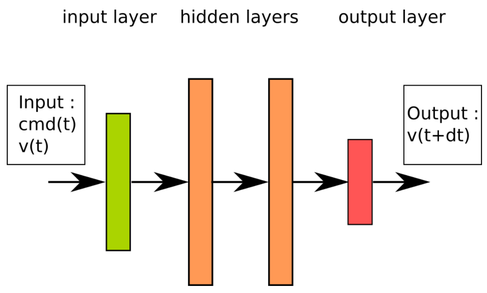


Figure 2: Neural Network

ARX and NN comparison

To evaluate neural network model we compare the result of one step prediction with an ARX model.

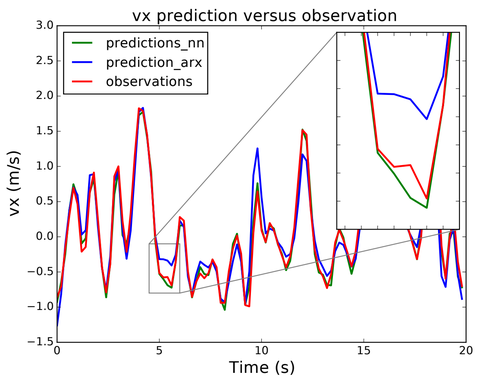


Figure 3: comparison of ARX and neural network model on linear dynamic.

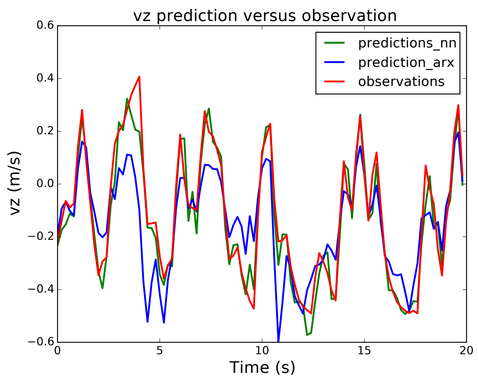


Figure 4: comparison of ARX and neural network model on non linear dynamic.

Data efficiency problems

NN are a powerful way to model system but its success depends on the quantity and quality of data.

Data prioritization

To address these problems we adapt the prioritized replay from [2]. The idea is that some samples are more important to the training than other thus we should focus the learning on these.

Algorithm

Calculating the importance of a sample is done by evaluating the model for each sample. The worse our model the more important the sample. To validate our approach we compare the multistep prediction error on the drone position for different model.

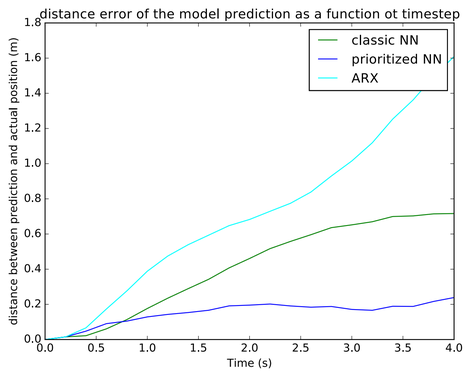


Figure 5: comparison of ARX, neural network and neural network with prioritization on multistep error.

Algorithm 1 Prioritized Sampling

Require: data,

$trainingData \leftarrow data$

$sampleWeight \leftarrow \emptyset$

$F \leftarrow Train(trainingData)$

N number of sample in $data$

for $i = 0$ to N **do**

$\delta_i \leftarrow \|Y_i - F(X_i, U_i)\|$

$P(i) \leftarrow \frac{\delta_i^\alpha}{\sum_k \delta_k^\alpha}$

$w_i \leftarrow \left(\frac{1}{N}\frac{1}{P(i)}\right)^\beta$

$sampleWeighted \leftarrow \{w_i\}_{0 \leq i \leq N}$

$trainingData \leftarrow$ sample data $d_i \sim P(i)$

end for

Experimental result

To evaluate the quality of the models in term of control we use them as part of an MPPI controller. We evaluate the controller in a trajectory following task.

Evaluation task

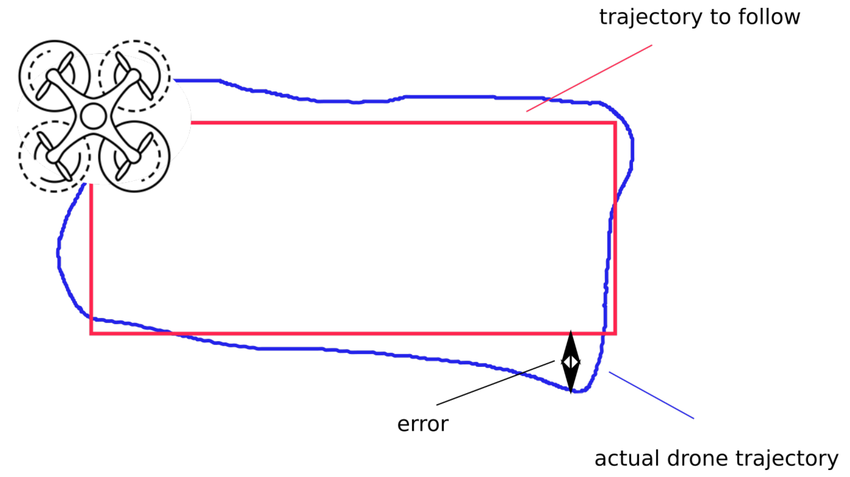


Figure 6: illustration of the evaluation task

One aspect of the machine learning method is its capacity to continuously adapt the model to the situation. Learning from the data collected while performing the task.

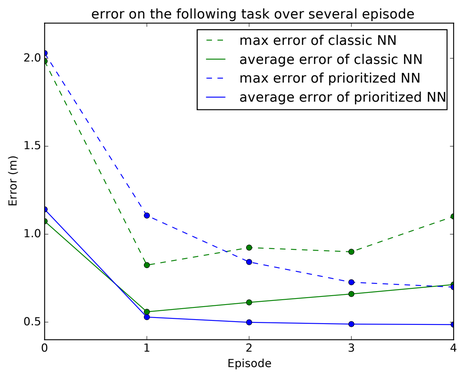


Figure 7: Neural Network versus Neural Network with prioritization.

We evaluate the controller on the task over several episodes, comparing the performance of the same neural network using normal training and the prioritized one.

Between each episode, the drone lands and the network is retrained on both previously and newly collected data.

The graph shows that the prioritized version trains more efficiently (learns faster) and is less sensible to unbalanced dataset.

Unbalanced drone problem

Finally to push the possibility of the method to the limit we modify the drone by adding a suspended mass to it, making it's dynamic much more complex. We then try to learn the new model of the modified drone.

The neural network learn enough of the dynamics for the controller to be able to do the task.

However it does not manage to improve it's performance after multiple trial.

Possible explanations are: the dynamic new complexity might require a new network architecture. It is also possible that the interaction of the low level controller and the mass makes the dynamic more stochastic, which would require a new approach.

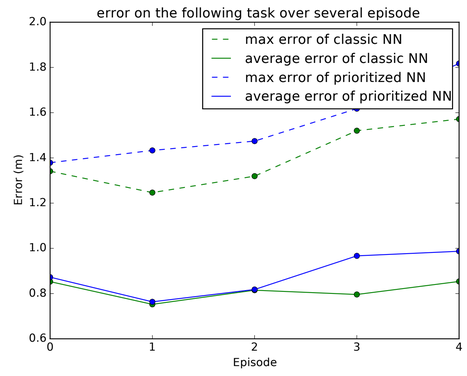


Figure 8: evaluation of the models on the unbalanced task

Forthcoming Research

The promising result open several opportunity for future research. The study of the meta-parameter of the prioritization is an interesting subject than might help to understand the condition that allow prioritization to work. The study of the possibility to adapt to changing condition could be useful for transfer learning and the capacity of such a scheme to adapt to new data quickly might be interesting.

References

[1] Lennart Ljung et al. Theory for the user. *Prentice Hall*, 1987.

[2] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[3] Bruno Scherrer, Mohammad Ghavamzadeh, Victor Gabillon, Boris Lesner, and Matthieu Geist. Approximate modified policy iteration and its application to the game of tetris. *J. Mach. Learn. Res.*, 16(1):1629–1676, January 2015.

[4] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. Information theoretic mpc for model-based reinforcement learning.

[5] T. Zhang, G. Kahn, S. Levine, and P. Abbeel. Learning Deep Control Policies for Autonomous Aerial Vehicles with MPC-Guided Policy Search. *ArXiv e-prints*, September 2015.

Acknowledgments

This work is done under the Grande Région rObotique aérienNE (GRoNe) project, funded by a European Union Grant through the FEDER INTERREG VA initiative.